PARAMETER ESTIMATION OF AN AUGMENTED AIRCRAFT USING NEURAL NETWORKS

A Thesis Submitted

in Partial Fulfilment of the Requirements

for the Degree of

MASTER OF TECHNOLOGY

by

G.V.S Bhaskar Reddy

to the

DEPARTMENT OF AEROSPACE ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

Dec, 1998

3 1 MAR 1999

CENTRAL LIBRARY
I. I. T., KANPUR

No. A 127817

A127817

POST GRADUATE OFFICE
I.I.T. Kanpur.
Submitted on 28.12.98

Certificate

It is certified the work contained in the thesis entitled "**Parameter Estimation of an Augmented Aircraft using Neural Networks** ", by G. V. S. Bhaskar Reddy , has been carried out under our supervision and that this work has not been submitted elsewhere for a degree.

Dr. S. C. Raisinghani
Professor
Department of Aerospace Engineering
I.I.T., Kanpur

Dr. A. K. Ghosh
Department of Aerospace Engineering
I.I.T., Kanpur

Acknowledgements

I express my deep sense of gratitude to my esteemed teachers and thesis supervisors, Dr S.C. Raisinghani and Dr. A.K. Gosh for their invaluable guidance, constructive criticism and persistent encouragement throughout this work. I shall always remain indebted to them for the precious time they have spared for me and the patience with which they always enlightened my path.

I express my special thanks to my friends Mr. Ch. Srinivasa Rao and Mr. P. N. Rao for their tremendous help in bringing this work to the present form.

I have no words to express my thanks to my parents and my family members who have been constant source of inspiration to me. I wish to thank all my friends and well wishers who made my stay at I.I.T, Kanpur memorable and pleasant.

Bhaskar Reddy G.V.S

Abstract

A new thrust is emerging in the area cf aircraft aerodynamic modelling and parameter estimation: development of techniques using artificial neural networks for flight vehicle identification. In the past, the most widely used parameter estimation methods have been equation error methods, output error methods and filter error methods. Applications of these methods require a priori postulations of an aircraft model. On the other hand, a class of neural networks called the feed forward neural networks (FFNNs) work as general function approximators, and are capable of approximating any continuous function to any desired accuracy. The present thesis deals with the modelling of an augmented aircraft, unstable in open-loop but made stable by simple controllers with constant gains in the feedback loops. For such an augmented aircraft, a recently proposed method called the Delta method using the FFNNs is utilized to estimate its open-loop stability and control derivatives (parameters) and the gains of the controllers present in the feedback loops. Results for simulated flight data, with and without measurement noise, and for different values of controller gains are presented. It is shown that the neural approach of modelling aircraft aerodynamics and subsequently using the neural model to estimate parameters and controller gains via the Delta method is an attractive alternative to the hitherto used conventional approaches.

Contents

1 Introduction 1

2 Feed Forward Neural Networks 6

2.1 Back-Propagation Algorithm 7

2.2 Network Parameters 8

2.3 Merits and Demerits of Neural Networks over Conventional methods for Aerodynamic modelling of an Augmented aircraft 11

3 Parameter Estimation Method and Generation of Simulated Flight Data 14

3.1 General 14

3.2 The Delta Method 14

3.3 Simulated Data Generation 15

3.4 Parameter Estimation using the Delta method 17

4 Results and Discussion 22

5 Conclusions 42

Bibliography 43

List of Figures

1.1 Schematic diagram of simulated neuron 2

1.2 Schematic of FFNN for longitudinal aerodynamic modelling . . 3

3.1 Block diagram of closed loop system 17

3.2 Schematic of Approach A 20

3.3 Schematic for estimating controller gains when w alone is fed back 20

3.4 Schematic of Approach B 20

3.5 Block diagram of closed loop system with w and q feed back . . 21

3.6 Schematic for estimating controller gains when both w and q are fed back 22

4.1 Comparison of pilot control input(δ_p) and elevator control input (δ_e) for w feed back 28

4.2 Different control inputs for generating simulated flight data . . . 29

4.3a Histograms for parameter estimates 30

4.3b Histograms for parameter estimates 31

4.4 Comparison of estimated responses with true responses 32

List of Tables

1 Comparison of estimated parameters via Approach A and Approach B for w feed back;FFNN architecture fixed 32

2 Comparison of estimated parameters via Approach A and Approach B; individual FFNN architecture 33

3 Comparison of estimated parameters via Approach A and Approach B for various number of iterations 34

4a Effect of measurement noise (1%) on parameter estimates via Approach A and Approach B 35

4b Effect of measurement noise (5%) on parameter estimates via Approach A and Approach B 36

5 Effect of different pilot control inputs on parameter estimates via Approach A and Approach B 37

6a Effect of w and q feed back on parameter estimates via Approach A and Approach B; $K_q = 0.15$; measurement noise = 0 38

6b Effect of w and q feed back on parameter estimates via Approach A and Approach B; $K_q = 0.15$; measurement noise = 5% 39

7a Effect of w and q feed back on parameter estimates via Approach A and Approach B; $K_q = 0.30$; measurement noise = 0 40

7b Effect of w and q feed back on parameter estimates via Approach A and Approach B; $K_q = 0.30$; measurement noise = 5% 41

List of Symbols

I_y	moment of inertia about y-axis, kg-m^2
K_p	constant used in shaping pilot input
K_q	gain constant in feed back loop for q
K_w	gain constant in feed back loop for w
L	total lift
M	pitching moment
X	computed output of neural network
Y	desired output of neural network
Δ	perturbation
δ_e	elevator deflection
δ_p	pilot command input
a_z	normal acceleration
m	aircraft mass, kg
q	pitch rate, rad/s
w	aircraft vertical velocity, m/s

Superscript:

$'$ equivalent derivatives

$\dot{}$ derivative with respect to time

Stability and Control derivatives:

$$L_w = \frac{1}{m}\frac{\partial L}{\partial w}$$

$$L_{\delta_e} = \frac{1}{m}\frac{\partial L}{\partial \delta_e}$$

$$M_w = \frac{1}{I_y}\frac{\partial M}{\partial w}$$

$$M_q = \frac{1}{I_y}\frac{\partial M}{\partial q}$$

$$M_{\delta_e} = \frac{1}{I_y}\frac{\partial M}{\partial \delta_e}$$

Abbreviations:

ANN Artificial Neural Network

FFNN Feed Forward Neural Network

MSE Mean Square Error

Chapter 1

Introduction

Recently, investigations have been carried out to explore the potential of artificial neural networks (ANNs) for aircraft aerodynamic modelling[1–4] and parameter estimation.[5–7] ANNs are inspired by the neuronal architecture of the human brain. These new computational models are massively adaptive systems that rely on surprisingly simple process units and arrangements of dense interconnections. ANNs can be best defined as, "an interconnected assembly of simple processing elements, units or nodes, whose functionality of the network is stored in the inter-unit connection strengths, or weights, obtained by a process of adaption to, or learning from, a set of training patterns". A simulated neuron or processing element in a neural network, such as that shown in Fig. 1.1 typically has many input paths and combines, usually by a simple summation, the values from these input paths. The combined input is then modified by a transfer function. The transfer function, also called the 'activation function', acts as a threshold function which only passes information if the combined activity level reaches a certain level. A neural network consists of many neurons or processing units joined together and is typically structured in the form of a sequence of layers. Each of the connections between neurons is assigned its individual weight and it is the adjustment of these weights which allows ANNs to act as computing devices, generating specific responses to a specific set of inputs. Out of many types of ANNs, it is the Feed Forward

Neural Networks (FFNNs) that have been found most promising for their application to aircraft modelling and parameter estimation.[5,7] The FFNN has neurons arranged in layers like directed graphs, implying unidirectional flow of signals, and thus are static in nature. There is an input buffer (layer) where data are presented to the network and an output buffer (layer) which holds the response of the network to a given set of input. Layers distinct from the input and the output layer are called the internal or hidden layers. It has been shown that the FFNNs can work as general function approximators and thereby are capable of approximating any continuous function to any desired accuracy provided the appropriate number of hidden layers and the neurons per layer exist, and that the activational function is continuous[8]. It is this ability of the FFNNs to act as general function approximators that FFNNs present themselves as an alternative tool for modelling aircraft aerodynamics.

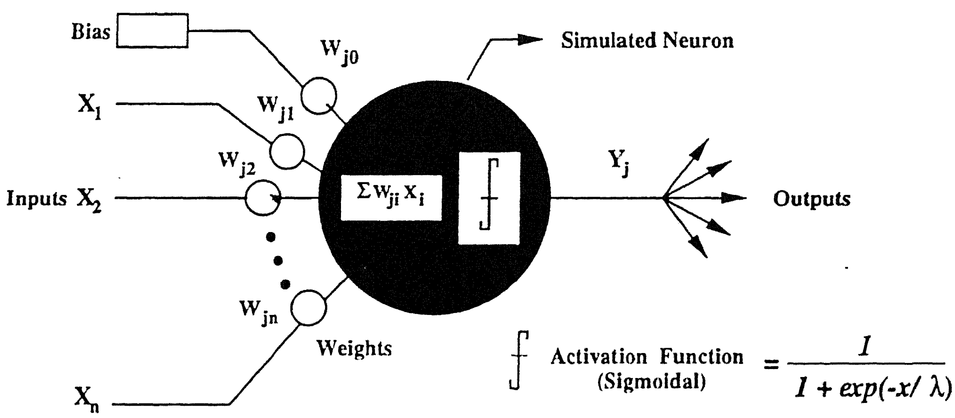


Fig. 1.1 Schematic diagram of a simulated neuron

For the purpose of longitudinal aerodynamic modelling using FFNN (Fig. 1.2), the network input variables are the longitudinal motion and control variables such as the vertical velocity (w), the pitch rate q, the elevator deflection (δ_e), etc. The output variable is the total lift L or the pitching moment M. During the supervised sequential (pattern) learning sessions of the network, the predicted values of L or M are compared with the corresponding known

(desired) values. The difference between the predicted and the known values yields an error at each time point; the errors are back propagated using the method called the Back propagation algorithm(BPA). The BPA essentially treats error function as a function of network weights, and uses an iterative descent gradient algorithm in the weight parameters space to minimize the mean square error (MSE) between the predicted and the known values of output variables. The iterative process is terminated when the MSE, defined below, is less than the prescribed value.

$$MSE = \frac{1}{m \times n} \sum_{j=1}^{n} \sum_{i=1}^{m} [Y_i(j) - X_i(j)]^2 \qquad (1.1)$$

where Y and X are, respectively, the desired and the predicted outputs; n is the number of data points; and m is the number of output variables.

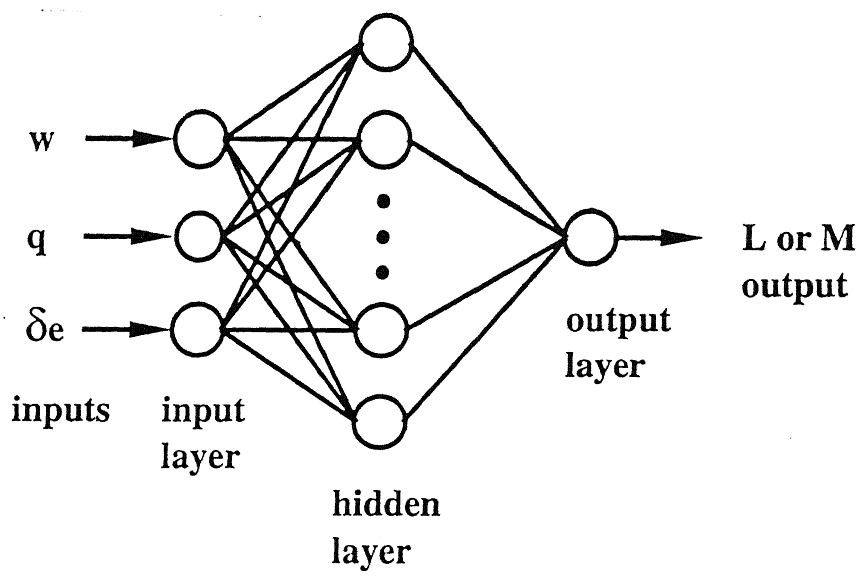


Fig. 1.2 Schematic of FFNN for longitudinal aerodynamic modelling

Raisinghani et al[5] and Ghosh et al[6] have recently shown that the neural models of aircraft aerodynamics present themselves as an attractive alternative for estimating aircraft parameters. They have proposed two new methods,

named the Delta method and the Zero method, successfully applied them to extract parameters from flight-data of a stable aircraft[5,6].

However, the need of high maneuverability has led to the design of many aerodynamically unstable aircraft configurations[9]. The introduction of these highly maneuverable, highly augmented and statically unstable aircraft present new problems in the field of system identification. In spite of progress made in the area of parameter estimation, the problem of estimating stability and control derivatives from flight data of a highly unstable system poses many difficulties till date. In a recent paper by Jategaonkar [9], an overview of a few recently introduced methods along with some of the conventional ones has been presented. The system identification of an aircraft unstable in open-loop, poses several practical difficulties owing to the following reasons:

1. The most widely used output-error methods, though in principle applicable to the unstable models, pose severe practical difficulties because they require the integration of the unstable equations of motion that may lead to a diverging solution owing to the numerical difficulties. For example, the inherent instabilities of the system equations propagate the growth of errors that may be introduced by poor initial values, round off or discretization.

2. The frequency-domain application of parameter estimation models like the maximum likelihood method[10,11] is possible for unstable aircraft except that the approach is limited only to linear equations of motion . Thus this mathematical limitation prevents this method to be used for many modern day fighter aircrafts that cover extreme flight regimes where they exhibit nonlinear aerodynamic characteristics.

3. An unstable aircraft is made stable in close-loop, and the controller designed for the purpose tends to suppress the oscillations and transient motions, and thereby, reducing the information content in the measured flight data.

4. There is limitation of degradation in the accuracy of estimation because of correlations between the input and the output variables caused by the presence of feed-back loops in the flight control system.

A critical appraisal of the available methods[9] indicates that a need exists to find a better and more efficient method for parametric estimation from flight data of an unstable aircraft.

In the present thesis, an augmented aircraft, unstable in open-loop but made stable by simple controllers with constant gains in the feed-back loop is considered. For such an augmented aircraft, it is shown that the recently proposed Delta method using the FFNNs can be advantageously applied to estimate its open-loop stability and control derivatives (parameters) as well as the gains of the controllers present in the feed-back loops.

The work carried out in the thesis is presented in 5 separate chapters. In chapter 1, the present status of the FFNNs for aircraft aerodynamic modelling and parameter estimation is highlighted. The difficulties of parameter estimation of an unstable aircraft using the conventional methods is reviewed.

Chapter 2 presents a brief discussion about the FFNNs. The steps involved in the back propagation algorithm are also given. The effect of tuning parameters on the efficiency of training and prediction capabilities of the FFNN are highlighted. Finally, the relative merits of using FFNN for the parameter estimation of an augmented aircraft are pointed out.

The parameter estimation method and the procedure for generation of simulated flight data has been discussed in chapter 3.

In chapter 4, results for simulated flight data, for different values of controller gains, with and without measurement noise are presented.

The thesis is concluded in chapter 5, with a summary of the work, and possibilities of future applications and expansion of the work.

Chapter 2

Feed Forward Neural Networks

The feed forward neural networks are multi-layer neural networks with one or more layers of neurons between the input and output neurons. There are no interconnections between neurons in the same layer. However, each neuron's output in a layer provides an input to each of the neurons in the succeeding layer.

A typical three layer feed forward network with three input neurons, one hidden layer and one output layer is shown in Fig. 1.2. The number of neurons in the input and output layer are determined, respectively, by the number of input and output patterns, while the number of neurons in the hidden layer is decided by the complexity of the problem. The action of the hidden layer may be viewed as successive transformations of the original presentation in the input layer.

Many researchers suggest a rule of thumb that the number of neurons in the hidden layer must be large enough to address the problem and still be small enough such that the weights can be reliably estimated from the training data.

FFNNs are static and characterized by unidirectional flow of variable, i.e., the inputs are propagated through hidden layers to the output layers. Each

node in the layer (input or hidden layer) is connected to each node in the next layer (with the hidden or the output) through a connective weight. The connection weights are the variables that are dynamically adjusted to produce a given output. It is through such dynamic modification of a large number of variable weights from which the neural network derives much of the learning and cognitive power.

2.1 Back-Propagation Algorithm

Multi-layer neural networks are trained by supervised learning. One training algorithm in particular, the Back-propagation training algorithm was a major breakthrough and revived the popularity of multi-layer neural networks. The Back-propagation algorithm is a supervised adaptive learning algorithm popularized by Rumelhart et al., 1986. It minimizes the mean square error, given in Eq. (1.1), between the desired output and the computed output by adjusting the interconnecting weights using the gradient descent method.

The Back-propagation algorithm in brief can be described in the following steps:

1. Initialize all the weights and internal thresholds to small random values.
2. Present input vector $[x0, \ldots, xn]$ and corresponding output vector $[z0, \ldots, zn]$.
3. Calculate the actual output vector from the net, $[y0, \ldots, yn]$.
4. Using a recursive algorithm, start from the output nodes and work back to the first hidden layer and adjust the weights by

$$\Delta w_{ij} = \mu \delta_j x_i \qquad (2.1)$$

where w_{ij} is the weight from the i-th node in the lower layer to the j-th node in the upper layer; x_i is either the output of node i, or is an input; μ

represents a trial-independent learning rate; δ_j is an error term for node j. If the node j is an output node, then

$$\delta_j = y_j(1-y_i)(z_j - y_j) \tag{2.2}$$

where z_j is the desired output of node j and y_j is the actual output. If node j, is an internal node, then

$$\delta_j = x_j(1-x_j)\sum \delta_k w_{jk} \tag{2.3}$$

where k is over all nodes in the layer right above the node j.

5. Repeat by going to step 2.

6. Stop when the change in weight is below a certain predefined criterion, i.e., when the mean square error is less than the prescribed value.

After the training process, the network is believed to have learned the underlying relationship between the input and the output vectors through a final set of connection weights which minimizes the overall global error. So, in a recall process, also called the prediction phase, if a previously unseen input is presented to the network, the network will produce an output in accordance with the relationship induced by the training. This is taken as a reasonable solution for the specific problem.

2.2 Network Parameters

There are many influencing tuning parameters of the FFNN that affect the training and thereby prediction capabilities of the network. In any practical application of back propagation algorithm, its level of efficiency depends on the network parameters. A proper selection of the network parameters is necessary for the given set of training data. Unfortunately there are no definite rules for selection of these network parameters, and yet, one has to find the best possible set of parameters, even if by trial and error. The MSE as defined

in the eqn. (1.1) has been the beacon for selecting the optimum network architecture. For the aerodynamic modelling, the following influencing parameters were considered:

1. Number of hidden layers
2. Initial network weights
3. Number of hidden nodes
4. Learning rate parameter
5. Momentum rate parameter
6. Number of Iterations
7. Logistic gain or slope factor

A matrix of tuning parameters was generated wherein each parameter was varied within a prescribed range, and the network was trained for various combinations of tuning parameters to arrive at the best possible set that led to the minimum MSE for the given data set. Main findings of the study are briefly described below:

Effect of the number of hidden layers, k

To study the effect of the hidden layers on the training, the modelling of the selected data was carried out with only one hidden layer, and then, with two hidden layers. In both the cases the predicted output matched well with the desired output. Thus, the results suggested that both types of networks provide adequate representation of the model being sought. To save computational time, it was therefore decided that the FFNN with only one hidden layer is adequate for all the investigations carried out and reported in the thesis.

Effect of initial network weights, $\mathbf{W}_{ij}$

The initial weights are generally set to small random values. The algorithm

does not work well if the initial weights are either zero or poorly chosen non-zero values. It was observed that the smaller initial random weights within the range of -0.5 to 0.5 were adequate for good convergence.

Effect of the number of neurons in the hidden layer

The number of neurons in the hidden layers is decided by the complexity of the problem. The number of connective weights increases with the increase in the number of nodes. An arbitrary increase in the number of the nodes makes the network architecture more complex, increases the training time and makes the training more difficult. For the flight data analyzed, 2 to 6 nodes in the hidden layers were found adequate. Any increase in the number of nodes beyond 6 did not make any significant contribution towards improvement of training, rather it only costs more in terms of the training time.

Effect of the learning parameter, μ

The Learning rate parameter μ controls the update of the connection weights, and this in turn, controls the rate of convergence. A proper learning rate helps the network to converge fast. A lower value of learning rate results in slow convergence. On the other hand, if the learning rate is large, the learning becomes unstable; the net oscillates back and forth across the error minimum. Learning rates in the range of 0.6 to 0.90 were found to be appropriate for most of the flight data used for training the network.

Effect of the momentum parameter, Ω

The momentum parameter is also known as the acceleration term, since it helps to accelerate the convergence. The momentum parameter has the tendency to smooth out small fluctuations in the error-weight space (it acts like a low-pass filter). In combination with other parameters, its value in the range of 0.70 to 0.90 was found to be the most appropriate.

Effect of the number of iterations, I

Generally, it was observed that an increase in the number of iterations led to a rapid decrease of MSE up to a certain number of iterations. For most of the

sets of flight-data, it was observed that the number of iterations in the range of 2000-5000 was sufficient to arrive at an adequate model satisfying the MSE criterion.

Effect of the Logistic gain parameter, λ

The slope of the sigmoidal function changes with the value of logistic gain parameter. This parameter plays an important role in the convergence of the network. Values of λ within the range of 0.70 to 0.95 worked well for the FFNN modelling.

2.3 Merits and Demerits of Neural Networks over Conventional methods for Aerodynamic modelling of an Augmented aircraft

The conventional methods of parameter estimation (reviewed earlier) have one limitation: they need the a priori fix of a functional form of aerodynamic force and moment coefficients. Generally, the aerodynamic coefficients are assumed to be linear, polynomial, or spline function of the unknown parameters. Such an a priori fix of aerodynamic model converts model identification problem into a parameter estimation problem, and thus imposes severe restriction on the accuracy and validity of the resulting model. Such an approach would be susceptible to errors in dealing with large-amplitude, high angle of attack, and time dependent maneuvers. In contrast, the FFNNs can be regarded as a non-parametric modelling method, both the structure and parameters need not be known a priori. Thus the limitation of assuming specific form for aerodynamic coefficients is relaxed. This capability of FFNNs for representing aerodynamic model of an aircraft without the need of a formal model structure formulation has potential uses to perform real-time (on line) modelling. With the availability of high-speed processors and commercially, off-the-shelf neural network hardware, real-time simulation is possible. A FFNN model can be developed in real-time during a flight-test and the model can be downloaded

CENTRAL LIBRARY
I. I. T.
No. A 127817

into real-time during a simulation for pilot training. Such a real-time simulator will be specifically useful during aircraft full scale development, flight-test and evaluation program. Also, the neural networks are robust in the presence of measurement noise.

In conventional methods, the goal of modelling is intrepretability which favors structured models. When using FFNNs, intrepretability is sacrificed to achieve modelling of complex systems.

Thus in contrast to the conventional approach of aircraft parameter estimation based on the well understood basic principles underlying the aircraft dynamics and aerodynamic forces and moments, the neural network approach leads to a black-box model to which no physical significance can be attributed, either to its structure or weights.

When compared to the conventional methods, the most important drawback of many artificial neural network models is a lack of perspicuity. Their massive parallelism, nonlinearities, and adaptive characteristics all conspire to render analytic treatment of artificial neural networks fairly limited in scope. The selection of network architecture for neural models suffer from two major shortcomings: 1) the space of possible artificial neural network architectures is extremely large and it is simply impractical to evaluate a reasonable variety of architectures, 2) what constitutes a good architecture is intimately dependent on the application; further more, the architecture needs to be separately determined for the specific problem under consideration, for which at present we have no techniques or methods available in the literature.

Apart from the problem of optimal network architecture design, the neural modelling also suffers from three major drawbacks discussed below:

Convergence: One major problem in neural modelling is about the training times in practical applications. There is no proof that the networks will converge with a finite step size. Empirical observations show that

networks usually train, but the duration of the training process is unpredictable and lengthy.

Local minima: The neural models that use the gradient descent methods adjust their weights following the local slope of the error surface toward a minimum. This works well with convex error surfaces, which have a unique minimum, but it often leads to non optimal solutions with the highly convoluted, non convex surfaces encountered in practical problems. In some cases, a local minimum is an acceptable solution; in others, it is inadequate. Even after the network has trained, there is no way to tell that the network has found the global minimum. If a solution is not satisfactory, one is obliged to shake the weights by initializing them to new random values and to retrain the network, with no guarantee that it will train on a given trial or that a global minimum will ever be found.

Paralysis: Under some circumstances, a network can train itself into a state in which weight modification comes to a virtual standstill. This "network paralysis" is a serious problem, once entered, it can extend training time by orders of magnitude. Also there is no theory that predicts whether or not a network will become paralyzed during training. The cost of paralysis can be high. Simulations can consume many hours of computer time, only to end in a paralytic training failure.

However, in the present thesis we did not encounter the problems of convergence and paralysis. The length of the training process was of the order of few minutes. However, the problem of local minima did arise on a few occasions wherein the weights were shaken by initializing them to new random values; the network was then retrained. This did overcome the problem of local minima.

Chapter 3

Parameter Estimation Method and Generation of Simulated Flight Data

3.1 General

In this chapter, we first describe the salient features of Delta-method[5] used for parameter estimation from flight data using FFNNs. Next we outline the procedure used and types of simulated data generated for analysis. Details of the example aircraft used and the types of control inputs utilized for generating simulated flight data are also given. Flight data with and without pseudo measurement noise were prepared for analysis.

3.2 The Delta Method

The Delta method[5] exploits the basic definition of what a stability or control derivative stands for: the stability/control derivatives represent the variation in the aerodynamic force or moment caused by a small variation in one of the motion/control variables about its nominal value, whereas all of the other variables are held constant. For example, dimensional derivative $M_{\dot{w}}$ represents

variation in pitching moment (M) with respect to vertical velocity (w), whereas all other variables like the pitch rate (q), elevator deflection (δ_e), etc., are held constant. To estimate M_w via the Delta method, the FFNN is first trained to map the network input file variables w, q, δ_e to the output file variable M (assumption: M is a function of w, q and δ_e only). Next, a modified network input file is prepared wherein w values at each time point are perturbed by $\pm\Delta$ w while all the other variables retain their original values. This modified file is now presented to the trained network and the corresponding predicted values of the perturbed M (M^+ for w+Δw and M^- for w -Δw) are obtained at the output node. The estimate for the derivative M_w is given by $M_w = (M^+ - M^-)/2\Delta w$. For predicting M^+ and M^- values, the delta method utilizes the generalization properties of the FFNN that allow the network to interpolate or extrapolate, as necessary, to predict output corresponding to inputs that are slightly different from the ones used for training the network. Because the histograms show that the parameter estimates have a near-normal distribution, the mean is taken as the estimated value and the sample standard deviation as the measure of its accuracy.

3.3 Simulated Data Generation

Because real flight data for an augmented aircraft were not available, simulated flight data are generated for an example aircraft operating in close loop. Figure 3.1 shows a block diagram of such a close loop system. In Fig. 3.1, K_w is the gain constant in feedback loop for w, and K_p is the constant used in shaping of the pilot elevator input. The feedback of w makes the unstable open-loop aircraft stable in close loop.

For numerical validation, simulated flight data for an example aircraft similar to Beaver aircraft are generated for the short period mode for a multi step 3-2-1-1 pilot input (δ_p). The data sets were generated for a range of values of K_w. Specifically, a fourth-order Runge-Kutta method was employed to

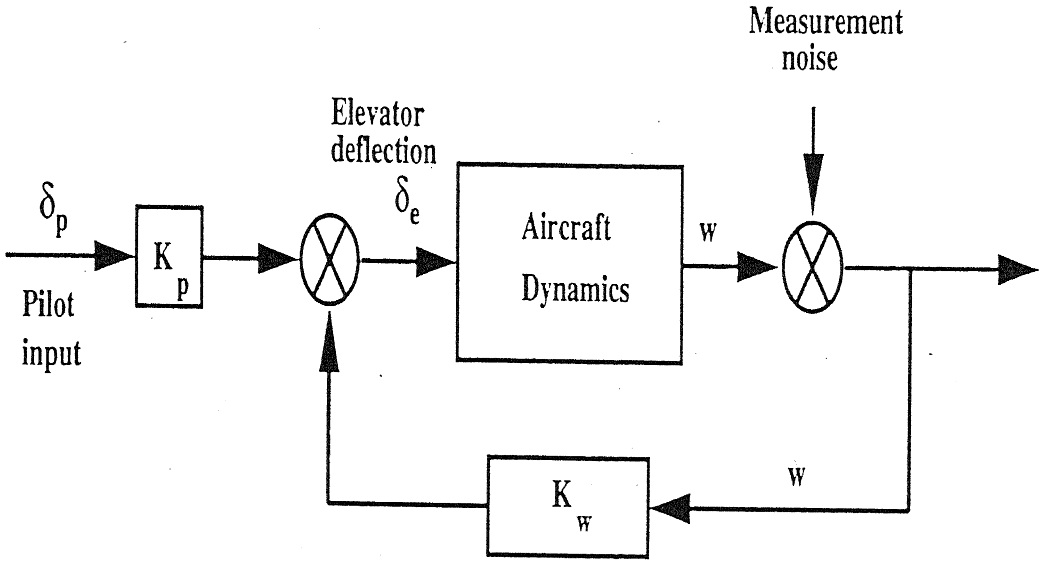


Fig. 3.1 Block diagram of closed loop system

numerically integrate the following equations:

$$\dot{w} - u_o q \quad = \quad -L \quad = \quad -L_w w - L_{\delta_e}\delta_e \qquad (3.1a)$$

$$\dot{q} \quad = \quad M \quad = \quad M_w w + M_q q + M_{\delta_e}\delta_e \qquad (3.1b)$$

$$\delta_e \quad = \quad K_p\delta_p + K_w w \qquad (3.1c)$$

where aircraft velocity $u_o = 50$m/s, and the dimensional derivatives parameters L_w, L_{δ_e}, ..., M_{δ_e} are assigned the values as given in Table 1. The value of K_w is varied from 0.025 to 1.0 and the value of K_p is fixed at 0.8. It may be emphasized that, for the case of real flight data, there is no need to either postulate or integrate Eqs. (3.1). For the real flight data, the measured w, q, and δ_e would form the network input variables; the network output variable L would be obtained using the measured vertical acceleration a_z, and M is

obtainable either using the measured $\dot{q}$, or from $\dot{q}$ computed via numerical differentiation of measured q. Thus for real flight data, L = -a_z and M=$\dot{q}$. For the present study, Eq. (3.1) has been solely used for the purpose of generating simulated flight data.

The FFNN is trained using the (w, q, δ_e) in the input file, and L or M in the output file. For each data set, a neural network architecture was searched for minimum MSE and high correlation coefficient. The search involved varying the network tuning parameters like the number of neurons in the hidden layers (2-6), the learning rate (0.2-0.8), the momentum rate (0.2-0.8), the logistic gain (0.8-0.95), the number of iterations (500-10000), etc. The best architecture so obtained is used first to train the network and subsequently for estimating parameters via the Delta method.

3.4 Parameter Estimation using the Delta method

The parameters to be estimated are the dimensional derivatives L_w, L_{δ_e}, M_w, M_q and M_{δ_e}, and the gain constants K_w and K_p. Two possibles approaches were attempted.

Approach A

If we substitute Eq. (3.1c) into (3.1a,b), the resulting equations will show the equivalent derivatives for the close loop system,

$$\dot{w} - qu_o = -L'_w\, w - L'_{\delta_e}\delta_p \qquad (3.2a)$$

$$\dot{q} = M'_w\, w + M_q\, q + M'_{\delta_e}\delta_p \qquad (3.2b)$$

where the prime derivatives on the right hand side of the equations are the equivalent derivatives defined as follows:

$$
\begin{aligned}
L'_w &= L_w + K_w L_{\delta_e}; \quad L'_{\delta_e} = K_p L_{\delta_e} \\
M'_w &= M_w + K_w M_{\delta_e}; \quad M'_{\delta_e} = K_p M_{\delta_e}
\end{aligned}
$$

To estimate equivalent derivatives via the Delta method, the network uses w, q, and δ_p as the inputs, and L or M as the output (Fig. 3.2). To retrieve the open-loop derivatives from the estimates of equivalent derivatives, we need the estimates of the gain constants K_w and K_p. To this purpose, the neural network is trained using the measured δ_p and w as the network inputs and the measured δ_e as the network output(Fig. 3.3). Application of the Delta method would now yield estimates of K_w and K_p as may be seen from Eq. (3.1c) and Fig. 3.3. For example, to estimate the controller gain K_w via the Delta method, the FFNN is first trained to map the network input file variables w, δ_p to the output file variable δ_e (assumption: δ_e is a function of w and δ_p, Eq. (3.1c)) as shown in Fig. 3.3. Next, a modified network input file is prepared wherein the w values at each time point are perturbed by $\pm\Delta w$ while the δ_p retains its original value. This modified file is now presented to the trained network and the predicted values of the perturbed δ_e are obtained at the output node. Say, a perturbation in w is selected to yield the predicted δ_e^+ for w + Δw and δ_e^- for w - Δw. The controller gain K_w is now given by $K_w = \frac{\delta_e^+ - \delta_e^-}{2(\Delta w)}$. Similarly, by perturbating δ_p, the controller gain K_p is given by $K_p = \frac{\delta_e^+ - \delta_e^-}{2(\Delta \delta_p)}$.

Approach B

The network is trained using the measured w, q and δ_e as inputs and L or M as the output (Fig. 3.4). In this case, the Delta method will directly yield the open-loop derivatives. The gain constants are separately estimated as described for approach A.

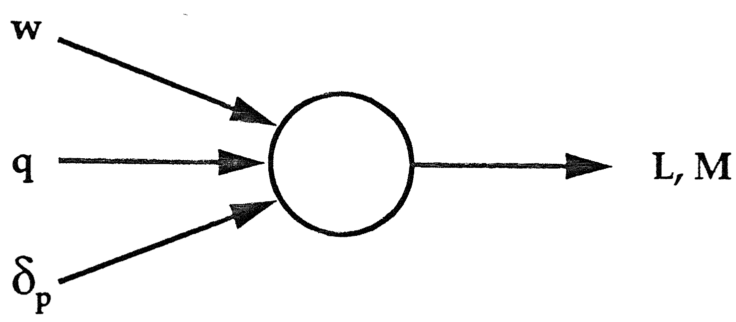


Fig. 3.2 Schematic for Approach A

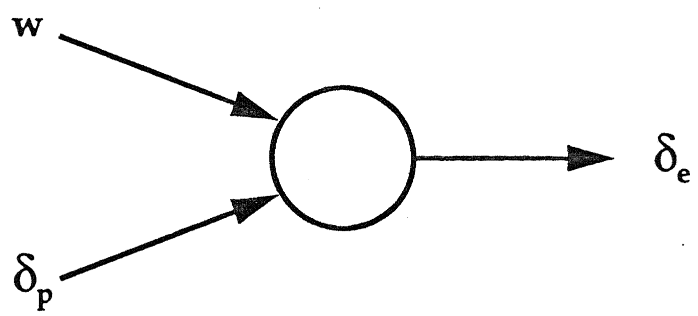


Fig. 3.3 Schematic for estimating controller gain

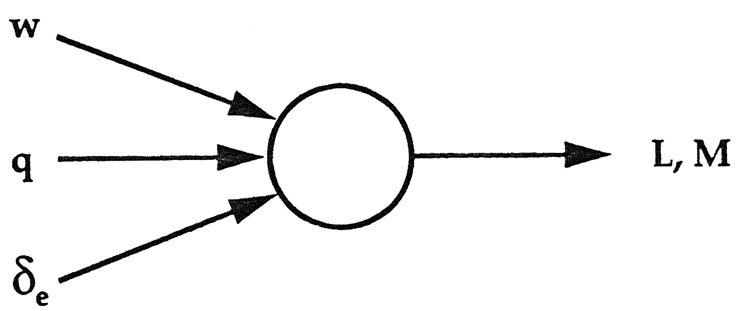


Fig. 3.4 Schematic for Approach B

In the due course of the thesis work it was realized that an augmented airplane may have multiple feed back loops, and it was of interest to see how

addition of another feed back loop would affect the parameter estimates. To this purpose, q was also fed back to essentially increase the damping ratio of the short period mode. Fig. 3.5 shows the block diagram of such a close-loop system. In Fig. 3.5, K_q is the controller gain in the feed-back loop for q feed back and K_w and K_p are the gain constants defined previously.

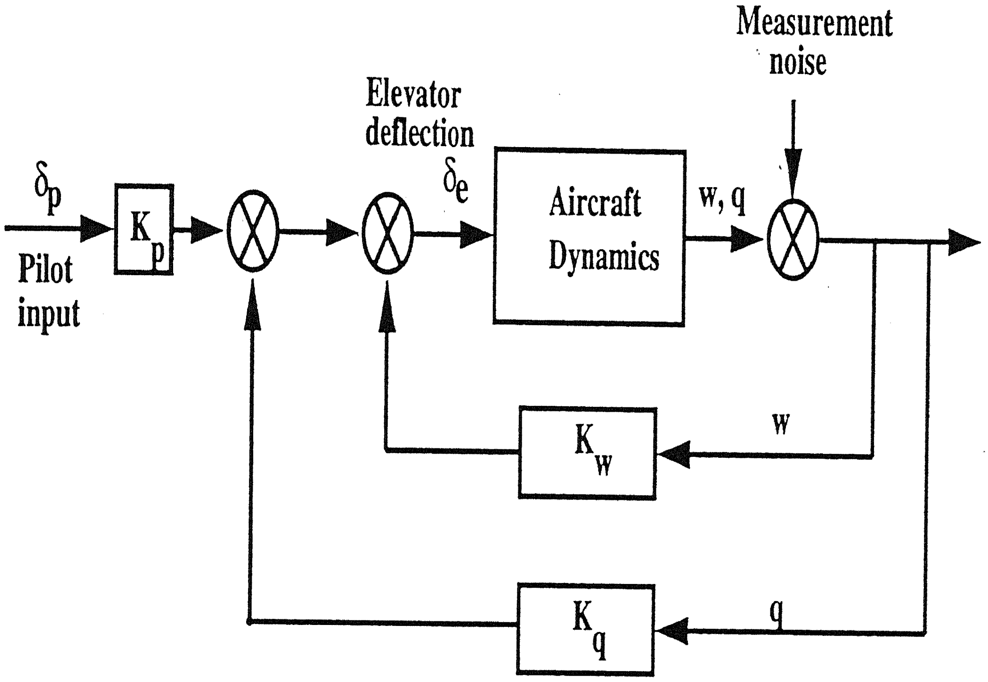


Fig. 3.5 Block diagram of closed loop system with w and q feed back

Due to the feed-back of q, Eqs. (3.1) get modified to the following form:

$$\dot{w} - u_o q = -L = -L_w w - L_{\delta e}\delta e \qquad (3.3a)$$

$$\dot{q} = M = M_w w + M_q q + M_{\delta e}\delta e \qquad (3.3b)$$

$$\delta e = K_p\delta_p + K_w w + K_q q \qquad (3.3c)$$

Now for the parameter estimation of the dimensional derivatives L_w, $L_{\delta e}$, M_w, M_q, and $M_{\delta e}$ and the gain constants K_p, K_w, and K_q from the Approach

A, we substitute Eq. (3.3c) into Eq. (3.3a,b), the resulting equations will show the equivalent derivatives for the close loop system:

$$\dot{w} - qu_o = -L'_w w - L'_q q - L'_{\delta_e}\delta_p \qquad (3.4a)$$

$$\dot{q} = M'_w w + M'_q q + M'_{\delta_e}\delta_p \qquad (3.4b)$$

where the prime derivatives on the right hand side of the above equations are the equivalent derivatives. The equivalent derivatives are defined as follows:

$$L'_w = L_w + K_w L_{\delta_e}; \quad L'_q = K_q L_{\delta_e}; \quad L'_{\delta_e} = K_p L_{\delta_e}$$

$$M'_w = M_w + K_w M_{\delta_e}; \quad M'_q = M_q + K_q M_{\delta_e}; \quad M'_{\delta_e} = K_p M_{\delta_e}$$

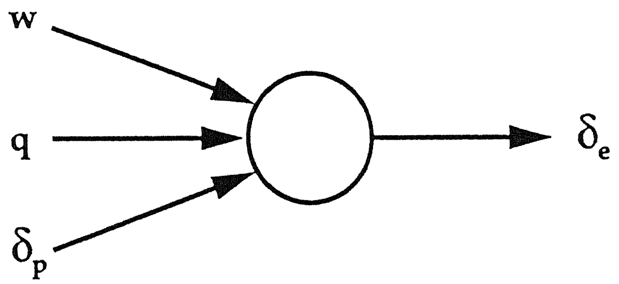


Fig. 3.6 Schematic for estimation of controller gains when w and q are fed back

Now the equivalent derivatives are estimated via Delta method as defined earlier in Approach A (Fig. 3.2). To retrieve the open-loop derivatives from the estimates from the equivalent derivatives, the estimates of the gain constants are needed. To this purpose, the neural network is trained using the δp, w and q as the network inputs, and the measured δe as the network output (Fig. 3.6). Application of the Delta method would yield estimates of K_p, K_w, and K_q as may be seen from Eq. 3.3c and Fig. 3.6.

Chapter 4

Results and Discussion

In this chapter, parameter estimates of an augmented aircraft via both the approaches A and B for K_w values ranging from 0.025 to 0.50, K_q values for 0.15 and 0.30, and the value of K_p fixed at 0.8 are presented. The effects of the number of iterations, and the use of individual optimal neural network architecture on the accuracy of the estimates are also discussed. Results for flight data with and without pseudo measurement noise are presented for both the approaches A and B.

Table 1 shows results for varying values of K_w with K_p fixed at 0.8 and the feed back loop having only one controller gain K_w for the w feed back. As a first attempt, the optimal neural network architecture for $K_w = 0.025$ was found, and freezing this particular network architecture,the estimates for other values of K_w were estimated for a quick study. Results for Approach A and Approach B are given in Table 1; the results for different values of K_w in the range of 0.025 to 0.50 are compared with true the values.

As seen from Table 1, estimates via approach B are marginally superior to the approach A. The approach B results indicate that all the stability and control derivatives, and the controller gains are estimated accurately, and with a high level of confidence as indicated by low sample standard deviations. For most of the combinations of K_w (0.025 to 0.50) values, the estimated parame-

ters compared well with the true values; the only exception being the case of $K_w = 0.05$. There seemed no obvious reason for estimates to be relatively less accurate for $K_w = 0.05$ in comparison to the rest of the values tried for K_w. However, one possible explanation for it could be as follows: the parameters for $K_w \neq 0.025$ were estimated using the optimal network architecture found for specific data for $K_w = 0.025$. It was, therefore, conjectured that the estimates in case of $K_w = 0.05$ could be improved if its own optimal network architecture was used. To this purpose, individual optimal architecture was searched for two typical values of $K_w = 0.05$ and 0.5 to see if it leads to any significant improvement in the estimated values. The results for $K_w = 0.05$ and 0.50 are presented in Table 2.

From Table 2, it can be seen that the accuracy of parameter estimates for $K_w = 0.05$ and 0.50 does improve significantly. Especially, the estimates corresponding to $K_w = 0.05$ have improved significantly when compared to its earlier estimates given in Table 1. It is thus concluded, that for every new set of input and output patterns that characterize a given problem, the data specific optimal architecture has to be found for the purpose of accurate estimates. In view of this, all further studies were carried out by finding individual optimal architecture for the data set being analyzed. The results presented hereafter are those obtained by using such optimal architecture for the specific cases discussed.

Effect of Iterations on the parameter estimates

Next, a study was undertaken to asses the effect of the number of iterations (I) on the accuracy of the estimated parameter. The architecture of the neural network was fixed while the number of iterations were varied from 500 to 10,000. As mentioned in chapter 2, the neural network could be trained satisfactorily within 500 iterations. However, parameter estimation via the Delta method showed that an increase in the number of iterations up to 5,000 did lead to better estimates with a rapid decrease in the mean square error. How-

ever, beyond 5,000 iterations, the improvement was only marginal. In Table 3, the parameter estimates from simulated flight-data via Approach A and Approach B are compared for the case of $K_w = 0.05$ for 500, 5000 and 10,000 iterations. It can seen from Table 3 that, for 5,000 iterations, the estimates are superior than for 500 iterations; this is reflected in terms of the mean values being closer to the true values as well as the sample standard deviation being much lower than those for 500 iterations. Also the mean square error for 5,000 iterations is low when compared to that for 500 iterations. As seen from Table 3, the improvement in accuracy of estimates is only marginal when number of iterations are increased from 5,000 to 10,000. In order to save the computational time most of the simulated flight-data in this thesis are analyzed for 5000 iterations only.

Effect of measurement noise on parameter estimates

Next, a study was undertaken to see the effect of measurement noise on the parameter estimates. In real life situations, the recorded flight-data is generally corrupted with measurement noise. In order to investigate the effects of measurement noise on parameter estimates, simulated pseudo noise of varying intensities was added to the motion variables (w, q) and the aerodynamic forces (L, M). The reason for adding noise to L and M are as follows. In the case of real flight-data, we would have access to noisy data of w, q, a_z and $\dot{q}$, along with control input δ_e which is generally assumed to be noise free. Conversion of noisy a_z and $\dot{q}$ to L and M will yield to noisy L and M. It is thus appropriate to add noise to L and M. The noise was simulated by generating successively uncorrelated pseudo random numbers having normal distribution with zero mean and assigned standard deviation, the standard deviation corresponds approximately to designated percentage (1% and 5%) of the maximum amplitude of the corresponding variable. Results from both the approaches A and B when applied to 1 % and 5 % noisy data are compared with true values in Table 4a and Table 4b respectively.

From Table 4a, it can be seen that the parameter estimates for 1% noise

do not deteriorate much in comparison with the true values. Even for case of 5% noise (Table 4b), the estimates are only marginally inferior. It may be observed that the estimates from the Approach B are marginally superior than for the Approach A for both the cases of 1% and 5% noise levels. The results suggest that the Approach B is more robust than the Approach A with respect to measurement noise.

Effect of different pilot control inputs on parameter estimates

Although, the parameter estimates for multistep 3-2-1-1 pilot control input (δ_p) are reasonably good, but the actual control input acting on the aircraft is the elevator control input(δ_e) which gets modified due to the presence of feed back in the close loop system as shown in Fig. 4.1. From Fig. 4.1, it can be seen that the elevator control input(δ_e), which acts on the aircraft, no longer retains the 3-2-1-1 form but gets modified into an arbitrary form; the form acquired depends on the value of feed back gain K_w. This suggested that a need exists to search for optimal control inputs for close loop systems[12]

This motivated us to study the effect of different pilot control inputs on parameter estimates. To this purpose, simulated flight data were generated for different pilot control inputs; a few of the control inputs tried are shown in Fig. 4.2. The results for these pilot control inputs are presented in Table 5. From Table 5, it can be seen that the estimates for the multistep 3-2-1-1 δ_p are superior as compared to the rest of the δ_p inputs of Fig. 4.2.

Effect of w and q on parameter estimates

The results are now presented for the aircraft augmented with two(w and q) feed back loops(Fig. 3.5). The results for both approach A and B are presented in Table 6 and 7. Results for $K_p = 0.8$, $K_w = 0.05$ and 0.50, $K_q = 0.15$ are given for no noise in Table 6a, and for 5% noise in Table 6b. For the same gain constants and measurement noise, but for $K_q = 0.30$, the corresponding results are given in Table 7a(no noise) and Table 7b(5% noise).

From Table 6 and 7, it can be seen that the estimates via the approach B

are marginally superior to the Approach A. The Approach B results indicate that all the stability and control derivatives as well as the controller gains are estimated accurately, and with a high level of confidence as indicated by low sample standard deviations. Similar levels of accuracy for parameter estimates were observed for all the combinations of controller gains used to generated data for the present study; the only exception being the case of $K_w = 0.05$. The relatively poor estimates for the case of $K_w = 0.05$ was initially quite puzzling and there seemed no obvious reason for such a behavior. However, on reflection, it was conjectured that the explanation for it may lie in the way δ_p is modified by a specific value of the controller gain K_w. The value of $K_w = 0.05$ seems to modify δ_p such that the resulting δ_e acting on the aircraft is not able to excite the short period mode as well as it was for other values of K_w. Thus, information content is reduced in the response, and thereby, the observed reduction in the identifiability of the system. This reinforces our earlier contention that a need exists to search for and design of optimal control inputs for close loop systems.[12] To illustrate a typical distribution of parameter estimates, Fig. 4.3a, b shows histograms of estimates corresponding to flight data for $K_w = 0.5$ in Table 6a. The figure shows that the estimates have a near-normal distribution, and thereby the use of mean values as the estimated value and the sample standard deviation as the measure of accuracy.

Another way to illustrate the quality of parameter estimates is to compare the true responses (w and q) with the estimated response of the aircraft for the same pilot control input; the estimated response for both Approaches A and B is obtained by substituting estimated values of derivatives and controller gains into Eq. (3.3) and solving it for w and q. Figure 4.4 shows the comparison of true response with the estimated responses for Approach A and B for the case of $k_w = 0.5$ and measurement noise = 5%. The estimated response w and q for the both approaches A and B almost overlap in Fig. 4.4 and can not be distinguished on the scale of the figure. Between the estimated w and q responses, it may be noticed that the matching with the true responses is relatively better for q as compared to w. Similar matches between the true

and estimated responses were found for all the cases studied, and as in Fig. 4.4, the match for w were always marginally inferior to that for the q.

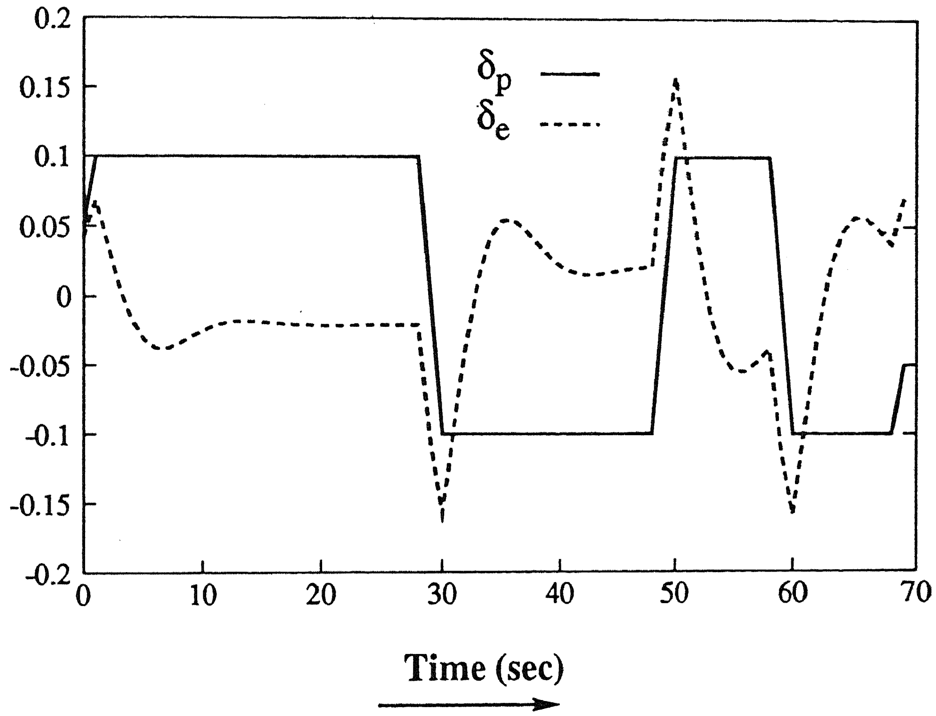


Fig. 4.1 Comparison of pilot control input(δ_p) and elevator control input(δ_e) for w feeback; $K_w = 0.05$ and $K_p = 0.80$.

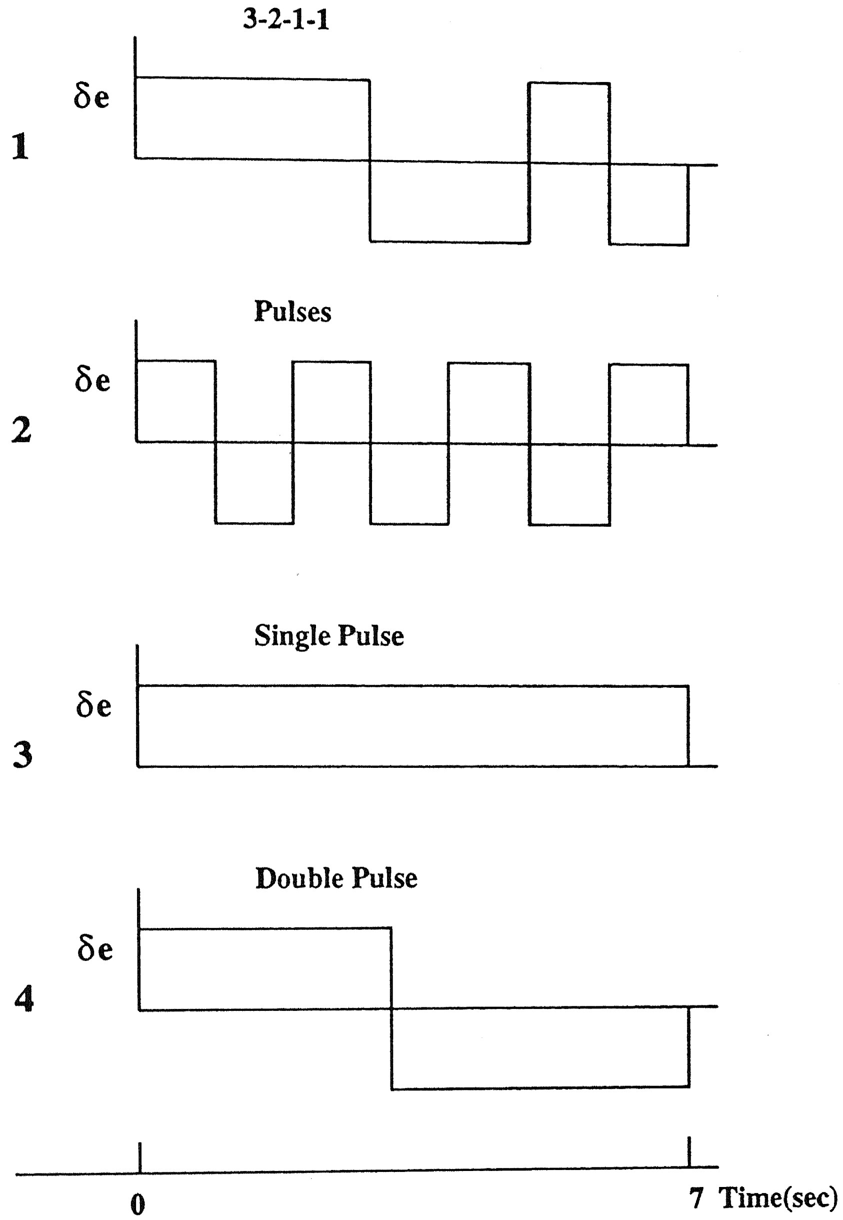


Fig. 4.2 The control inputs (1 to 4) used for generating simulated flight-data

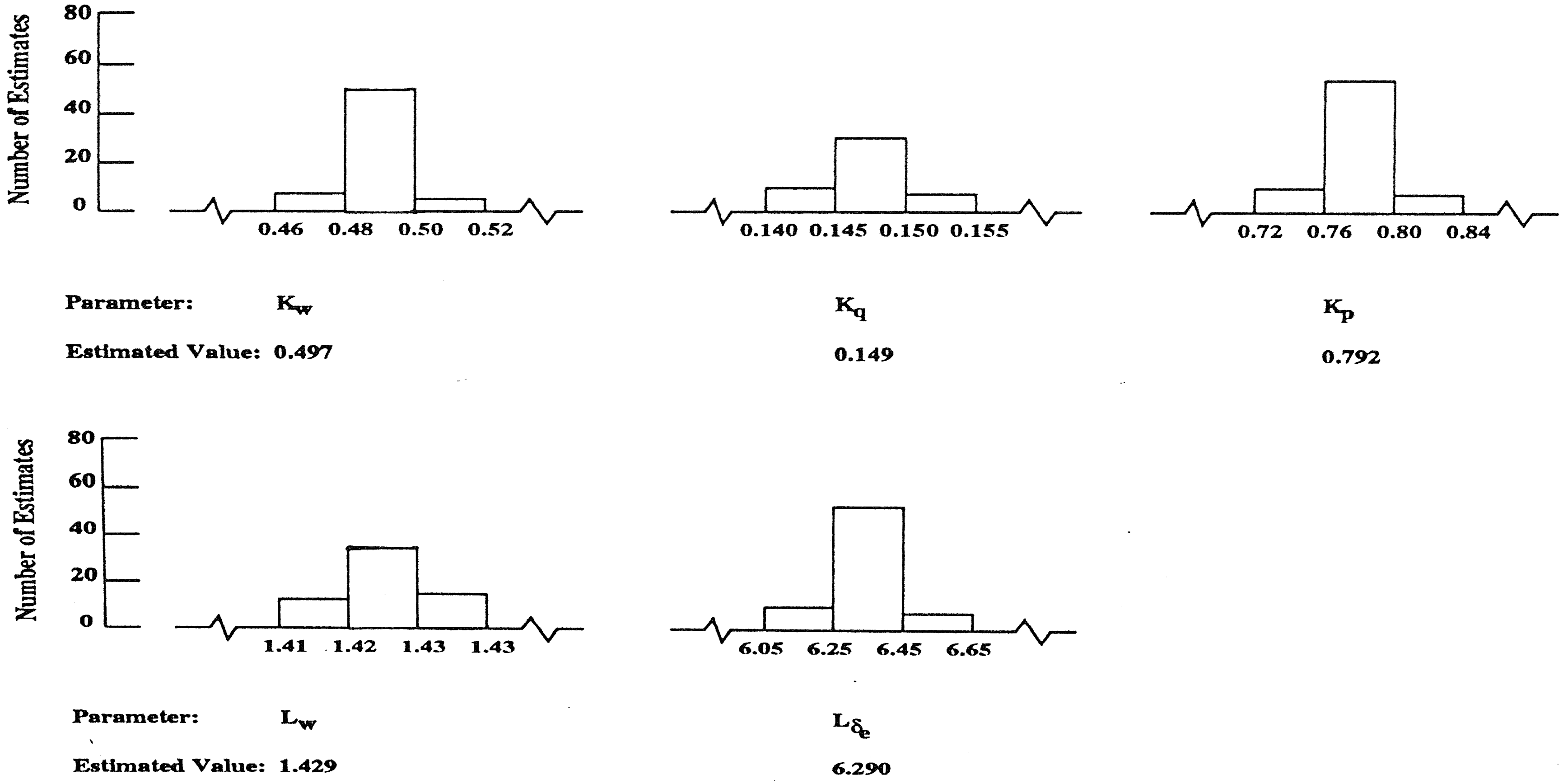
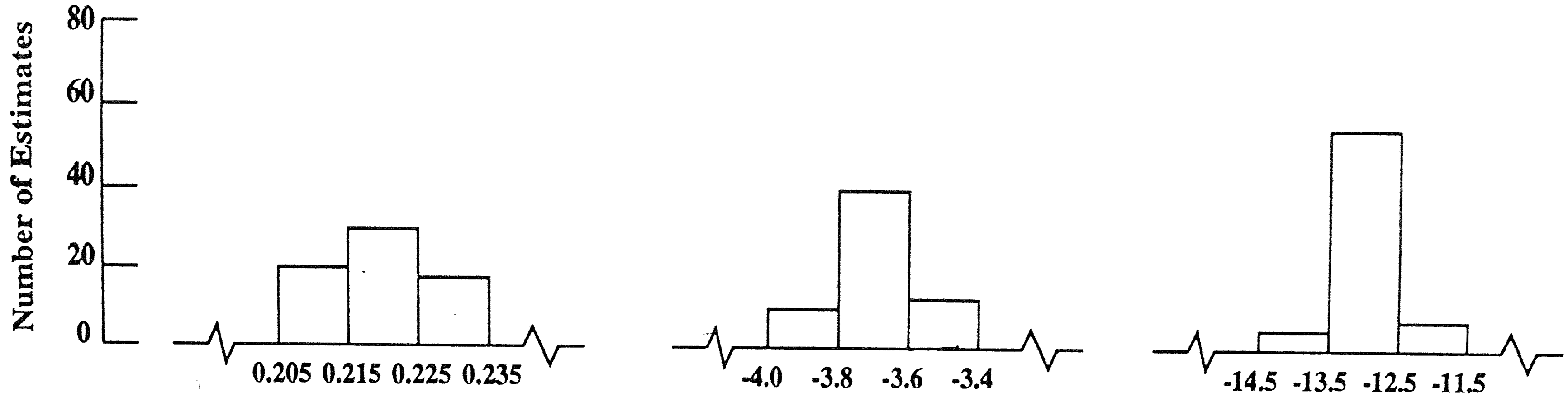


Fig. 4.3a Histograms for parameter estimates; $K_w = 0.50$; $K_q = 0.15$; $K_p = 0.80$; measurement noise = 0; Approach B



Parameter:	M_w	M_q	$M_{\delta e}$
Estimated Value:	0.222	-3.718	-13.057

Fig. 4.3b Histograms for parameter estimates; $K_w = 0.50$; $K_q = 0.15$; $K_p = 0.80$; measurement noise = 0; Approach B

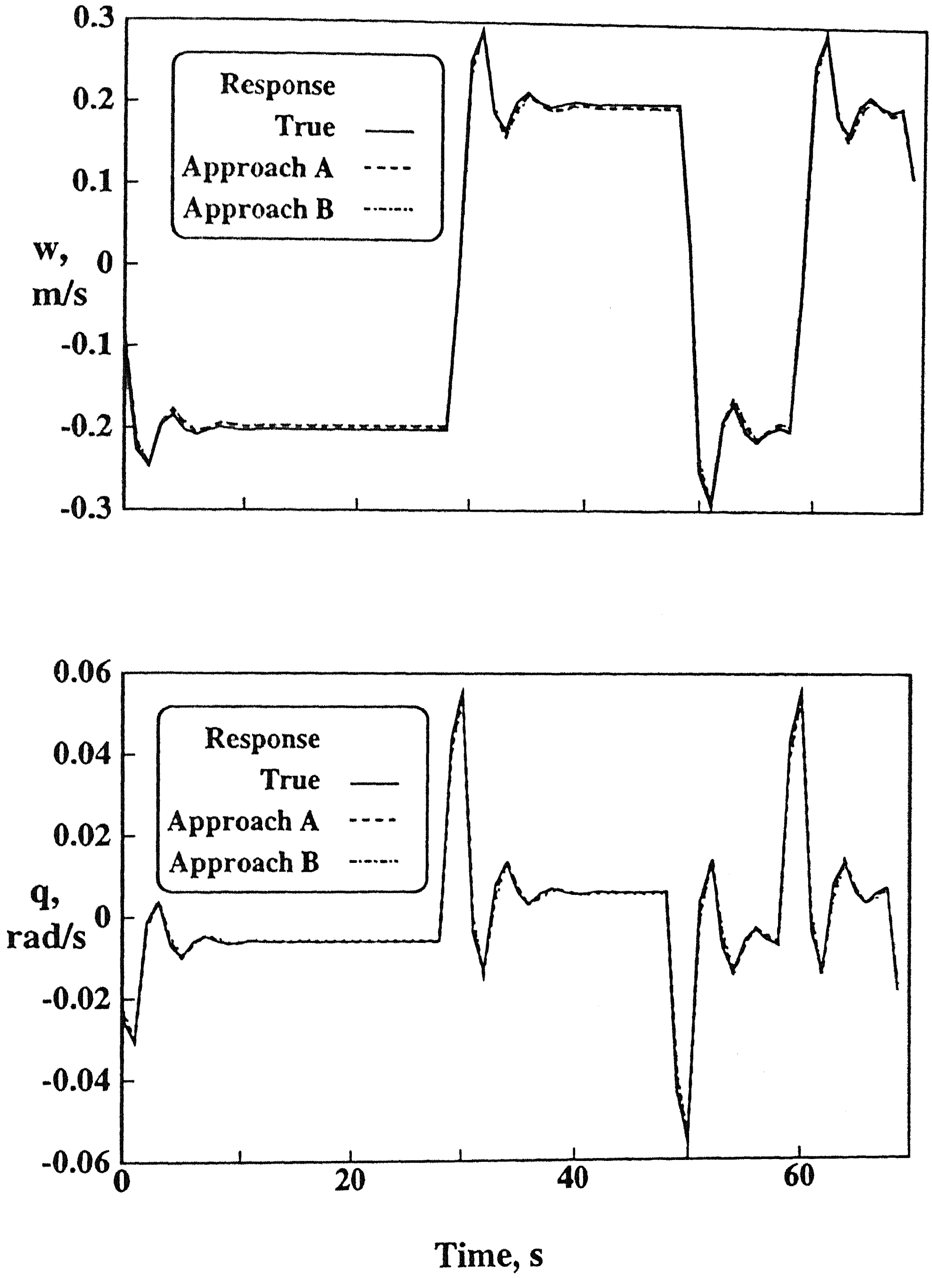


Fig. 4.4 Comparison of estimated responses with true response for $K_w = 0.5$, $K_q = 0.15$, $K_p = 0.8$, Noise = 5%.

Table 1 Comparison of estimated parameters via Approach A and Approach B for w feed back;FFNN architecture fixed

Para-meter	True Value	$K_w = 0.025$ A	$K_w = 0.025$ B	$K_w = 0.05$ A	$K_w = 0.05$ B	$K_w = 0.25$ A	$K_w = 0.25$ B	$K_w = 0.50$ A	$K_w = 0.50$ B
L_w	1.43	1.416 (0.086)†	1.427 (0.050)	1.485 (0.062)	1.465 (0.064)	1.392 (0.079)	1.431 (0.036)	1.427 (0.139)	1.432 (0.037)
$L_{\delta e}$	6.26	5.889 (0.325)	6.078 (0.196)	6.190 (0.336)	6.602 (0.304)	6.345 (0.163)	6.496 (0.171)	5.851 (0.556)	6.553 (0.167)
M_w	0.22	0.228 (0.005)	0.217 (0.005)	0.180 (0.033)	0.186 (0.017)	0.195 (0.076)	0.218 (0.006)	0.216 (0.167)	0.218 (0.005)
M_q	-3.71	-3.435 (0.205)	-3.671 (0.005)	-4.177 (0.286)	-4.076 (0.017)	-3.780 (0.095)	-3.709 (0.006)	-4.077 (0.101)	-3.811 (0.005)
$M_{\delta e}$	-12.8	-12.998 (0.559)	-12.697 (0.328)	-12.839 (0.820)	-12.928 (1.058)	-12.70 (0.331)	-12.908 (0.343)	-12.947 (0.340)	-12.812 (0.328)
K_w	-	0.025 (0.001)	0.025 (0.001)	0.047 (0.002)	0.047 (0.002)	0.259 (0.008)	0.259 (0.008)	0.529 (0.017)	0.529 (0.017)
K_p	0.80	0.809 (0.021)	0.809 (0.021)	0.840 (0.047)	0.840 (0.047)	0.820 (0.025)	0.820 (0.025)	0.805 (0.022)	0.805 (0.022)

† Sample standard deviation
A Approach A
B Approach B

Table 2 Comparison of estimated parameters via Approach A and Approach B; individual FFNN architecture

		$K_w = 0.05$		$K_w = 0.5$	
Parameter	True Value	Approach A	Approach B	Approach A	Approach B
L_w	1.43	1.418 (0.053)†	1.451 (0.060)	1.440 (0.039)	1.434 (0.035)
$L_{\delta e}$	6.26	6.423 (0.137)	6.377 (0.267)	6.131 (0.152)	6.492 (0.192)
M_w	0.22	0.233 (0.008)	0.223 (0.006)	0.217 (0.005)	0.219 (0.005)
M_q	-3.71	-3.630 (0.141)	-3.759 (0.110)	-3.802 (0.065)	-3.833 (0.102)
$M_{\delta e}$	-12.8	-12.744 (0.566)	-12.985 (0.375)	-12.667 (0.314)	-12.879 (0.334)
K_w	-	0.050 (0.001)	0.050 (0.001)	0.507 (0.004)	0.507 (0.004)
K_p	0.80	0.796 (0.024)	0.796 (0.024)	0.799 (0.021)	0.799 (0.021)

† Sample standard deviation

Table 3 Comparison of estimated parameters via Approach A and Approach B for various number of iterations

		500		5000		10000	
Parameter	True Value	Approach A	Approach B	Approach A	Approach B	Approach A	Approach B
		(0.00978)@	(0.00840)	(0.00192)	(0.00157)	(0.00182)	(0.00148)
L_w	1.43	1.172 (0.044)†	1.437 (0.068)	1.418 (0.053)	1.451 (0.060)	1.424 (0.033)	1.450 (0.058)
$L_{\delta e}$	6.26	5.560 (0.382)	6.191 0.302)	6.423 (0.137)	6.377 (0.267)	6.419 (0.129)	6.362 (0.260)
M_w	0.22	0.23 (0.019)	0.213 (0.0265)	0.233 (0.008)	0.223 (0.006)	0.221 (0.005)	0.222 (0.006)
M_q	-3.71	-4.505 (0.216)	-3.504 (0.878)	-3.630 (0.141)	-3.759 (0.110)	-3.690 (0.101)	-3.747 (0.110)
$M_{\delta e}$	-12.8	-14.307 (2.364)	-12.218 (2.399)	-12.744 (0.566)	-12.985 (0.375)	-12.717 (0.466)	-12.937 (0.380)
K_w	-	0.053 (0.007)	0.053 (0.007)	0.050 (0.001)	0.050 (0.001)	0.050 (0.0009)	0.050 (0.0009)
K_p	0.80	0.853 (0.105)	0.853 (0.105)	0.796 (0.024)	0.796 (0.024)	0.798 (0.019)	0.798 (0.019)

† Sample standard deviation @ Mean square error

Table 4a Effect of measurement noise (1%) on parameter estimates via Approach A and Approach B

		$K_w = 0.05$		$K_w = 0.5$	
Parameter	True Value	Approach A	Approach B	Approach A	Approach B
L_w	1.43	1.419 (0.059)†	1.450 (0.060)	1.443 (0.041)	1.444 (0.035)
$L_{\delta e}$	6.26	6.516 (0.142)	6.325 (0.266)	6.219 (0.171)	6.502 (0.186)
M_w	0.22	0.229 (0.009)	0.221 (0.006)	0.210 (0.007)	0.221 (0.008)
M_q	-3.71	-3.518 (0.152)	-3.742 (0.107)	-3.840 (0.076)	-3.783 (0.093)
$M_{\delta e}$	-12.8	-12.721 (0.596)	-12.942 (0.364)	-12.743 (0.325)	-13.05 (0.357)
K_w	-	0.050 (0.001)	0.050 (0.001)	0.494 (0.007)	0.494 (0.007)
K_p	0.80	0.793 (0.023)	0.793 (0.023)	0.791 (0.005)	0.791 (0.005)

† Sample standard deviation

Table 4b Effect of measurement noise (5%) on parameter estimates via Approach A and Approach B

		$K_w = 0.05$		$K_w = 0.5$	
Parameter	True Value	Approach A	Approach B	Approach A	Approach B
L_w	1.43	1.428 (0.079)†	1.445 (0.057)	1.374 (0.049)	1.394 (0.037)
$L_{\delta e}$	6.26	6.029 (0.312)	6.153 (0.260)	6.729 (0.181)	6.656 (0.167)
M_w	0.22	0.214 (0.019)	0.231 (0.010)	0.230 (0.011)	0.216 (0.006)
M_q	-3.71	-4.18 (0.152)	-4.07 (0.107)	-3.992 (0.076)	-3.769 (0.093)
$M_{\delta e}$	-12.8	-13.229 (0.712)	-13.158 (0.663)	-12.929 (0.394)	-12.993 (0.350)
K_w	-	0.049 (0.001)	0.049 (0.001)	0.514 (0.004)	0.514 (0.004)
K_p	0.80	0.772 (0.019)	0.772 (0.019)	0.787 (0.013)	0.787 (0.013)

† Sample standard deviation

Table 5 Effect of different pilot control inputs on parameter estimates via Approach A and Approach B

		3-2-1-1		Pulses		Single pulse		Double pulse	
Parameter	True Value	A	B	A	B	A	B	A	B
L_w	1.43	1.418 (0.053)†	1.451 (0.060)	1.339 (0.080)	1.386 (0.085)	1.374 (0.0001)	1.539 (0.038)	1.403 (0.084)	1.442 (0.081)
$L_{\delta e}$	6.26	6.423 (0.137)	6.377 (0.267)	5.883 (0.304)	5.927 (0.381)	5.378 (0.002)	6.26 (0.154)	5.975 (0.089)	6.444 (0.327)
M_w	0.22	0.233 (0.008)	0.223 (0.006)	0.241 (0.014)	0.216 (0.008)	0.177 (0.0001)	0.306 (0.007)	0.238 (0.012)	0.221 (0.025)
M_q	-3.71	-3.630 (0.141)	-3.759 (0.110)	-3.623 (0.577)	-3.719 (0.167)	-4.599 (0.113)	-4.15 (0.102)	-4.01 (0.341)	-3.857 (0.366)
$M_{\delta e}$	-12.8	-12.744 (0.566)	-12.985 (0.375)	-12.507 (1.550)	-12.914 (0.528)	-14.164 (0.002)	-15.057 (0.371)	-13.390 (0.709)	-13.166 (0.491)
K_w	-	0.050 (0.001)	0.050 (0.001)	0.0508 (0.001)	0.0508 (0.001)	0.0481 (0.001)	0.0481 (0.001)	0.519 (0.001)	0.519 (0.001)
K_p	0.80	0.796 (0.024)	0.796 (0.024)	0.803 (0.021)	0.803 (0.021)	0.875 (0.024)	0.875 (0.024)	0.829 (0.026)	0.829 (0.026)

† Sample standard deviation A Approach A B Approach B

Table 6a Effect of w and q feed back on parameter estimates via Approach A and Approach B; $K_q = 0.15$; measurement noise = 0

		$K_w = 0.05$		$K_w = 0.5$	
Parameter	True Value	Approach A	Approach B	Approach A	Approach B
L_w	1.43	1.319 (0.034)†	1.318 (0.044)	1.427 (0.018)	1.429 (0.037)
$L_{\delta e}$	6.26	5.303 (0.353)	5.690 (0.223)	6.520 (0.128)	6.290 (0.184)
M_w	0.22	0.210 (0.003)	0.225 (0.006)	0.232 (0.032)	0.222 (0.007)
M_q	-3.71	-3.561 (0.059)	-3.804 (0.095)	-3.722 (0.016)	-3.718 (0.093)
$M_{\delta e}$	-12.8	-12.668 (0.212)	-12.825 (0.349)	-13.164 (0.224)	-13.057 (0.340)
K_w	-	0.051 (0.001)	0.051 (0.001)	0.497 (0.013)	0.497 (0.013)
K_q	0.15	0.151 (0.004)	0.151 (0.004)	0.149 (0.004)	0.149 (0.004)
K_p	0.80	0.812 (0.021)	0.812 (0.021)	0.792 (0.002)	0.792 (0.002)

† Sample standard deviation

Table 6b Effect of w and q feed back on parameter estimates via Approach A and Approach B; K_q = 0.15; measurement noise = 5%

		$K_w = 0.05$		$K_w = 0.5$	
Parameter	True Value	Approach A	Approach B	Approach A	Approach B
L_w	1.43	1.318 (0.047)†	1.279 (0.063)	1.385 (0.013)	1.467 (0.039)
$L_{\delta e}$	6.26	5.361 (0.587)	5.301 (0.252)	6.595 (0.105)	6.492 (0.318)
M_w	0.22	0.213 (0.012)	0.235 (0.010)	0.176 (0.030)	0.188 (0.043)
M_q	-3.71	-3.626 (0.252)	-3.931 (0.189)	-4.256 (0.214)	-3.525 (0.192)
$M_{\delta e}$	-12.8	-12.017 (0.871)	-13.567 (0.658)	-11.729 (0.545)	-12.559 (0.428)
K_w	-	0.052 (0.001)	0.052 (0.001)	0.449 (0.012)	0.449 (0.012)
K_q	0.15	0.149 (0.003)	0.149 (0.003)	0.166 (0.004)	0.166 (0.004)
K_p	0.80	0.831 (0.020)	0.831 (0.020)	0.706 (0.019)	0.706 (0.019)

† Sample standard deviation

Table 7a Effect of w and q feed back on parameter estimates via Approach A and Approach B; K_q = 0.30; measurement noise = 0

		$K_w = 0.05$		$K_w = 0.5$	
Parameter	True Value	Approach A	Approach B	Approach A	Approach B
L_w	1.43	1.297 (0.065)†	1.315 (0.073)	1.388 (0.085)	1.440 (0.038)
$L_{\delta e}$	6.26	5.214 (0.363)	5.102 (0.353)	6.522 (0.048)	6.478 (0.163)
M_w	0.22	0.232 (0.010)	0.208 (0.013)	0.199 (0.098)	0.218 (0.005)
M_q	-3.71	-3.982 (0.121)	-3.758 (0.139)	-4.300 (0.353)	-3.827 (0.098)
$M_{\delta e}$	-12.8	-13.078 (0.551)	-12.856 (0.399)	-13.346 (0.878)	-13.086 (0.333)
K_w	-	0.052 (0.001)	0.052 (0.001)	0.499 (0.012)	0.499 (0.012)
K_q	0.30	0.315 (0.008)	0.315 (0.008)	0.304 (0.009)	0.304 (0.009)
K_p	0.80	0.839 (0.022)	0.839 (0.022)	0.795 (0.023)	0.795 (0.023)

† Sample standard deviation

Table 7b Effect of w and q feed back on parameter estimates via Approach A and Approach B; K_q = 0.30; measurement noise = 5%

		$K_w = 0.05$		$K_w = 0.5$	
Parameter	True Value	Approach A	Approach B	Approach A	Approach B
L_w	1.43	1.284 (0.072)†	1.308 (0.085)	1.368 (0.071)	1.495 (0.072)
$L_{\delta e}$	6.26	5.111 (0.372)	4.872 (0.412)	7.181 (0.372)	6.525 (0.331)
M_w	0.22	0.238 (0.012)	0.240 (0.0006)	0.250 (0.075)	0.212 (0.037)
M_q	-3.71	-3.400 (0.497)	-3.983 (0.104)	-4.289 (0.419)	-3.639 (0.090)
$M_{\delta e}$	-12.8	-11.343 (0.909)	-13.645 (0.345)	-12.876 (1.106)	-13.083 (0.366)
K_w	-	0.052 (0.001)	0.052 (0.001)	0.448 (0.012)	0.448 (0.012)
K_q	0.30	0.285 (0.007)	0.285 (0.007)	0.302 (0.008)	0.302 (0.008)
K_p	0.80	0.806 (0.020)	0.806 (0.020)	0.702 (0.019)	0.702 (0.019)

† Sample standard deviation

Chapter 5

Conclusions

Applicability of the recently proposed Delta method for estimating aircraft parameters from flight data using the FFNNs has been demonstrated for extracting open-loop parameters of an augmented aircraft operating in close loop configuration. The Delta method is quite straight forward in application for estimating parameters and controller gains of a fly-by-wire aircraft, the FFNN is trained on the measured flight data and the parameters/controller gains estimated in one try. The advantages of the FFNN based Delta method are : 1) no a priori math model of the aircraft needs to be postulated; 2) no guess need to be made of initial values of parameters, and thereafter solving the equations of motion to get the model response (as, for example, would be required by the ML method); 3) along with the stability and control derivatives, the controller gains are also estimated.

The present work has been limited to only simulated flight data and needs to be validated on real flight data—non-availability of such data was the handicap faced by us. Also, the present study considered only the short period dynamics and a simple control law (Eq. 3.3c) wherein the controller parameters are in the form of constant gains (K_w, K_q and K_p) only. Future directions of our work include modelling of more complex controllers and estimating its parameters. It is recommended for future research to seek real flight data and validate the proposed approaches A and B on such data.

Bibliography

[1] Hess, R. A., "On the Use of Back Propagation with Feed Forward Neural Networks for Aerodynamic Estimation Problem," AIAA Paper 93-3638, Aug. 1993.

[2] Youseff, H. M., "Estimation of Aerodynamic Coefficients Using Neural Networks," AIAA Paper 93-3639, Aug. 1993.

[3] Bassapa, and Jategaonkar, R. V., "Aspect of Feed Forward Neural Network Modelling and Its Application to Lateral-Directional Flight Data," DLR-IB 111-95/30, Braunschwieg, Germany, Sept. 1995.

[4] Linse, D. J., and Stengel, R. F., " Identification of Aerodynamic Coefficients Using Computational Neural Networks," Journal of Guidance, Control and Dynamics, Vol. 16, No. 6, 1993, pp. 1018-1025.

[5] Raisinghani, S. C., Ghosh, A. K., and Kalra, P. K., "Two New Techniques for Aircraft Parameter Estimation Using Neural Networks," The Aeronautical Journal, Vol. 102, No. 1011, 1998, pp. 25-29.

[6] Ghosh, A. K., Raisinghani, S. C., and Khubchandani, S., "Estimation of Aircraft Lateral-Directional Parameters Using Neural Networks," Journal of Aircraft, Vol. 35, No. 6, 1998, pp. 876-881.

[7] Raol, J. R., Jategoankar, R. V., "Aircraft Parameter Estimation Using Recurrent Neural Networks-A Critical Appraisal," AIAA Paper 95-3004, Aug. 1995.

[8] Hornik, K., Stinchcombe, M., and White, H., "Multi Layer Feed Forward Networks are Universal Approximators," Neural Networks, Vol. 2, No. 5, 1989, pp. 359-366.

[9] Jategoankar, R. V., and Thielecke, F., "Evaluation of Parameter Estimation Methods for Unstable Aircraft," Journal of Aircraft, Vol. 31, No. 3, 1993, pp. 510-519.

[10] Maine, R. E., and Iliff, K. W., "Identification of Dynamic Systems: Theory and Formulations," NASA RP 1138, Feb. 1985.

[11] Maine, R. E., and Iliff, K. W., "Application of Parameter Estimation to Aircraft Stability and Control - The Output-Error Approach," NASA RP 1168, Jan. 1986.

[12] Morelli, E. A., " Optimal Input Design for Closed Loop Modelling at High Angle of Attack," AIAA Paper 96-3418.